



PROIECT BAZE DE DATE NOSQL

Monitor Cheltuieli

Monitor Cheltuieli

In acest proiect, am utilizat MongoDB pentru a analiza si clasifica cheltuielile din colectia MonitorCheltuieli.

Andrei Hornaru

Introducere

Acest proiect isi propune sa analizeze cheltuielilor inregistrate in baza de date MongoDB, folosind operatii complexe de agregare pentru a extrage insight-uri valoroase. Bazandu-ne pe colectia MonitorCheltuieli, care contine inregistrari detaliate ale diferitelor tipuri de cheltuieli, ne propunem sa exploram si sa clasificam aceste date pentru a obtine o intelegere mai profunda a tendintelor si pattern-urilor de cheltuieli.

Prin utilizarea unor tehnici avansate de agregare, cum ar fi **\$group**, **\$match**, **\$project**, **\$unwind**, **\$bucket** si **\$facet**, am reusit sa realizam o serie de interogari care ne-au permis sa grupam, sa filtram si sa analizam datele din perspective multiple. Am aplicat **\$group** pentru a calcula sumele totale si medii ale cheltuielilor pe diferite categorii si locatii, si am folosit **\$match** pentru a filtra datele conform unor criterii specifice. De asemenea, am implementat **\$project** si **\$unwind** pentru a **descompune** si **restructura** datele, permitandu-ne sa examinam fiecare element al cheltuielilor in detaliu.

Un aspect central al analizei noastre a fost utilizarea operatorului **\$bucket** pentru a **clasifica** cheltuielile in categorii bazate pe suma lor, oferind o imagine clara asupra modului in care resursele financiare sunt alocate in diferite segmente. In plus, tehnica **\$facet** ne-a permis sa executam multiple pipeline-uri de **agregare simultan**, furnizand un set divers de rezultate intr-o singura interogare.

Prin acest proiect, am demonstrat capacitatea MongoDB de a gestiona si analiza seturi de date complexe, oferind insight-uri valoroase si ajutand la luarea unor decizii informate bazate pe date. Rezultatele obtinute sunt de o importanta majora pentru o gestionare eficienta a resurselor financiare si pentru planificarea bugetara.

Codul de creare colectie:

```
db.createCollection("MonitorCheltuieli", {
  validator: {
    "$jsonSchema": {
      "bsonType": "object",
      "properties": {
        "suma": {
          "description": "Suma cheltuielii",
          "bsonType": "double",
          "minimum": 0
        },
        "moneda": {
          "description": "Moneda sumei cheltuielii",
          "bsonType": "string",
          "minLength": 2
        },
        "data": {
          "description": "Data cheltuielii",
          "bsonType": "date"
        },
        "categoria": {
          "description": "Categoria cheltuielii",
          "bsonType": "string",
          "minLength": 2
        },
        "descriere": {
          "description": "Descriere cheltuiala",
          "bsonType": "string"
        },
        "locatia": {
          "description": "Locatia unde s-a efectuat cheltuiala",
          "bsonType": "string"
        },
        "documenteAsociate": {
          "description": "Documente asociate cu cheltuiala",
          "bsonType": "array",
          "items": {
            "description": "Referinta document",
            "bsonType": "string"
          },
          "minItems": 1 // 1 daca vrem sa fie obligatoriu un document atasat
        }
      },
      "required": ["suma", "moneda", "data", "categoria"],
      "additionalProperties": false
    }
  }
})
```

use MonitorCheltuieli

In aceasta comanda:

- suma, moneda, data, categoria sunt marcate ca fiind campuri obligatorii.
- descriere si locatia sunt optionale.
- additionalProperties este setat pe false pentru a nu permite alte proprietati decat cele definite.

Inserare date in colectie.

```
db.MonitorCheltuieli.insertMany([
  { suma: 248.26, moneda: "EUR", data: new ISODate("2023-01-03T00:00:00.000Z"),
    categoria: "Utilitati", descriere: "Factura electricitate", locatia: "Timisoara",
    documenteAsociate: ["Doc_1_1", "Doc_1_2"] },
  { suma: 492.38, moneda: "USD", data: new ISODate("2023-01-24T00:00:00.000Z"),
    categoria: "Divertisment", descriere: "Bilete la cinema", locatia: "Iasi", documenteAsociate:
    ["Doc_2_1", "Doc_2_2"] },
  { suma: 445.41, moneda: "RON", data: new ISODate("2023-01-01T00:00:00.000Z"),
    categoria: "Divertisment", descriere: "Bilet de autobuz", locatia: "Cluj", documenteAsociate:
    ["Doc_3_1", "Doc_3_2"] },
  { suma: 245.3, moneda: "EUR", data: new ISODate("2023-01-21T00:00:00.000Z"), categoria:
    "Educatie", descriere: "Bilete la cinema", locatia: "Iasi", documenteAsociate: ["Doc_4_1",
    "Doc_4_2"] },
  { suma: 258.1, moneda: "RON", data: new ISODate("2023-01-14T00:00:00.000Z"),
    categoria: "Utilitati", descriere: "Bilete la cinema", locatia: "Iasi", documenteAsociate:
    ["Doc_5_1", "Doc_5_2"] },
  { suma: 309.89, moneda: "RON", data: new ISODate("2023-02-02T00:00:00.000Z"),
    categoria: "Divertisment", descriere: "Bilete la cinema", locatia: "Timisoara",
    documenteAsociate: ["Doc_6_1", "Doc_6_2"] },
  { suma: 182.81, moneda: "USD", data: new ISODate("2023-02-28T00:00:00.000Z"),
    categoria: "Utilitati", descriere: "Bilete la cinema", locatia: "Cluj", documenteAsociate:
    ["Doc_7_1", "Doc_7_2"] },
  { suma: 129.47, moneda: "RON", data: new ISODate("2023-02-20T00:00:00.000Z"),
    categoria: "Educatie", descriere: "Bilet de autobuz", locatia: "Brasov", documenteAsociate:
    ["Doc_8_1", "Doc_8_2"] },
  { suma: 459.92, moneda: "RON", data: new ISODate("2023-02-02T00:00:00.000Z"),
    categoria: "Mancare", descriere: "Factura electricitate", locatia: "Iasi", documenteAsociate:
    ["Doc_9_1", "Doc_9_2"] },
  { suma: 131.77, moneda: "EUR", data: new ISODate("2023-02-12T00:00:00.000Z"),
    categoria: "Transport", descriere: "Factura electricitate", locatia: "Iasi", documenteAsociate:
    ["Doc_10_1", "Doc_10_2"] },
  { suma: 364.59, moneda: "USD", data: new ISODate("2023-03-19T00:00:00.000Z"),
    categoria: "Educatie", descriere: "Bilet de autobuz", locatia: "Brasov", documenteAsociate:
    ["Doc_11_1", "Doc_11_2"] },
```

```

    { suma: 360.07, moneda: "EUR", data: new ISODate("2023-03-14T00:00:00.000Z"),
      categoria: "Educatie", descriere: "Cina la restaurant", locatia: "Bucuresti",
      documenteAsociate: ["Doc_12_1", "Doc_12_2"] },
    { suma: 109.34, moneda: "EUR", data: new ISODate("2023-03-06T00:00:00.000Z"),
      categoria: "Transport", descriere: "Cursuri online", locatia: "Iasi", documenteAsociate:
      ["Doc_13_1", "Doc_13_2"] },
    { suma: 198.75, moneda: "EUR", data: new ISODate("2023-03-06T00:00:00.000Z"),
      categoria: "Mancare", descriere: "Factura electricitate", locatia: "Timisoara",
      documenteAsociate: ["Doc_14_1", "Doc_14_2"] },
    { suma: 235.68, moneda: "EUR", data: new ISODate("2023-03-26T00:00:00.000Z"),
      categoria: "Transport", descriere: "Bilete la cinema", locatia: "Cluj", documenteAsociate:
      ["Doc_15_1", "Doc_15_2"] }
  ])

```

Update:

```

db.MonitorCheltuieli.updateOne(
  { "_id": ObjectId("6557d08c1da4a56a043e4960") },
  { $set: { "descriere": "Cumpărături săptămânale" } }
)

```

1. Cheltuieli Mari: Sa se afiseze pentru fiecare document aferent cheltuielilor cu o suma mai mare de 300 de unitati: suma, moneda, data, categoria, descrierea si locatia.

```

db.MonitorCheltuieli.find({ "suma": { $gt: 300 } },
  { "suma": 1, "moneda": 1, "data": 1, "categoria": 1, "descriere": 1, "locatia": 1 })

```

```
MonitorCheltuieli | 0.192 s | 6 Docs
1 /* 1 createdAt:11/17/2023, 10:43:56 PM*/
2 {
3   "_id" : ObjectId("6557d08c1da4a56a043e4954"),
4   "suma" : 492.38,
5   "moneda" : "USD",
6   "data" : ISODate("2023-01-24T02:00:00.000+02:00"),
7   "categoria" : "Divertisment",
8   "descriere" : "Bilete la cinema",
9   "locatia" : "Iasi"
10 },
11
12 /* 2 createdAt:11/17/2023, 10:43:56 PM*/
13 {
14   "_id" : ObjectId("6557d08c1da4a56a043e4955"),
15   "suma" : 445.41,
16   "moneda" : "RON",
17   "data" : ISODate("2023-01-01T02:00:00.000+02:00"),
18   "categoria" : "Divertisment"
```

2. Cheltuieli in Categoria Divertisment: Sa se afiseze pentru fiecare document aferent cheltuielilor din categoria "Divertisment": suma, moneda, data, descrierea si locatia. Documentele vor fi afisate in ordine crescatoare in functie de suma.

```
db.MonitorCheltuieli.find({ "categoria": "Divertisment" },
    { "suma": 1, "moneda": 1, "data": 1, "descriere": 1, "locatia": 1 })
.sort({ "suma": 1 })
```

```
MonitorCheltuieli | 0.172 s | 3 Docs
1 /* 1 createdAt:11/17/2023, 10:43:56 PM*/
2 {
3   "_id" : ObjectId("6557d08c1da4a56a043e4961"),
4   "suma" : 235.68,
5   "moneda" : "EUR",
6   "data" : ISODate("2023-03-26T02:00:00.000+02:00"),
7   "descriere" : "Bilete la cinema",
8   "locatia" : "Cluj"
9 },
10
11 /* 2 createdAt:11/17/2023, 10:43:56 PM*/
12 {
13   "_id" : ObjectId("6557d08c1da4a56a043e4958"),
14   "suma" : 309.89,
15   "moneda" : "RON",
16   "data" : ISODate("2023-02-02T02:00:00.000+02:00"),
```

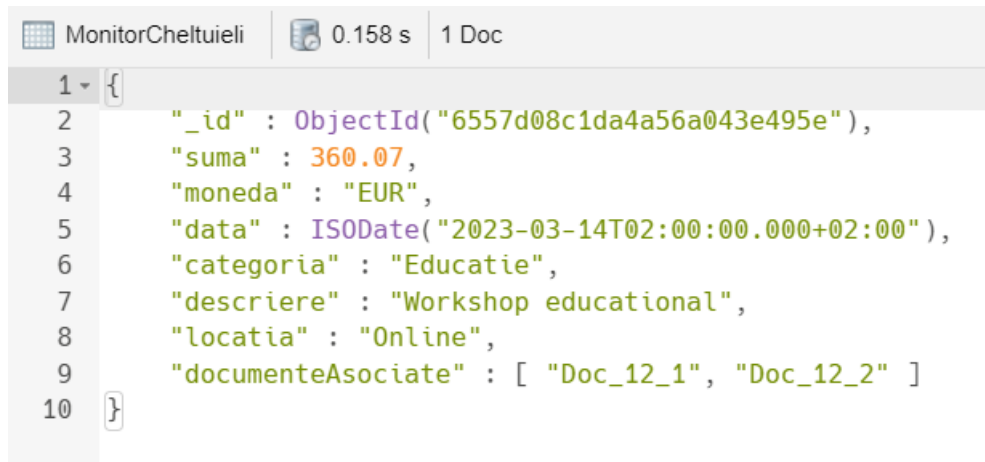
3. Cheltuieli cu Descrierea Specifica: Sa se afiseze cheltuielile care au in descriere cuvantul "cina". Se vor afisa sortate in functie de data cheltuielii.

```
db.MonitorCheltuieli.find(
  { "descriere": { $regex: /cina/, $options: 'i' } },
  { "suma": 1, "moneda": 1, "data": 1, "categoria": 1, "descriere": 1, "locatia": 1 }
).sort({ "data": 1 })
```

```
MonitorCheltuieli | 0.171 s | 1 Doc
1 {
2   "_id" : ObjectId("6557d08c1da4a56a043e495e"),
3   "suma" : 360.07,
4   "moneda" : "EUR",
5   "data" : ISODate("2023-03-14T02:00:00.000+02:00"),
6   "categoria" : "Educatie",
7   "descriere" : "Cina la restaurant",
8   "locatia" : "Bucuresti"
9 }
```

4. Actualizare locatiei acolo unde este "Bucuresti" se va inlocui cu valoarea "online".

```
db.MonitorCheltuieli.updateMany({ "locatia": "Bucuresti" },
    { $set: { "locatia": "Online" } })
```



The screenshot shows a MongoDB document in the 'MonitorCheltuieli' collection. The document is displayed in a code editor with line numbers 1 through 10. The document contains the following fields: '_id' (ObjectId), 'suma' (360.07), 'moneda' (EUR), 'data' (ISODate), 'categoria' (Educatie), 'descriere' (Workshop educational), 'locatia' (Online), and 'documenteAsociate' (an array containing 'Doc_12_1' and 'Doc_12_2').

```
1 {
2   "_id" : ObjectId( "6557d08c1da4a56a043e495e" ),
3   "suma" : 360.07,
4   "moneda" : "EUR",
5   "data" : ISODate( "2023-03-14T02:00:00.000+02:00" ),
6   "categoria" : "Educatie",
7   "descriere" : "Workshop educational",
8   "locatia" : "Online",
9   "documenteAsociate" : [ "Doc_12_1", "Doc_12_2" ]
10 }
```

5. Sa se afiseze toate documentele pentru cheltuielile care au in descriere termeni legati de "alimentatie" (de exemplu, "mancare", "restaurant", "pranz", "cina", etc.). Se vor afisa suma, data, descrierea si locatia.

```
db.MonitorCheltuieli.find(
{
  "descriere": {
    $regex: /bilet|cina/,
    $options: 'i' // optiunea 'i' face cautarea case-insensitive
  }
},
{
  "suma": 1,
  "data": 1,
  "descriere": 1,
  "locatia": 1
}
)
```

MonitorCheltuieli	0.163 s	5 Docs
1	/* 1 createdAt:11/17/2023, 10:43:56 PM*/	
2	{	
3	"_id" : ObjectId("6557d08c1da4a56a043e4954"),	
4	"suma" : 492.38,	
5	"data" : ISODate("2023-01-24T02:00:00.000+02:00"),	
6	"descriere" : "Bilete la cinema",	
7	"locatia" : "Iasi"	
8	},	
9		
10	/* 2 createdAt:11/17/2023, 10:43:56 PM*/	
11	{	
12	"_id" : ObjectId("6557d08c1da4a56a043e4955"),	
13	"suma" : 445.41,	
14	"data" : ISODate("2023-01-01T02:00:00.000+02:00"),	
15	"descriere" : "Bilet de autobuz",	
16	"locatia" : "Cluj"	

Aggregate functions:

6. Calculul sumei totale a cheltuielilor în fiecare moneda:

```
db.MonitorCheltuieli.aggregate([
  {
    $group: {
      _id: "$moneda",
      total: { $sum: "$suma" }
    }
  }
]);
```

```

1  /* 1 */
2  {
3      "_id" : "RON",
4      "total" : 1602.79
5  },
6
7  /* 2 */
8  {
9      "_id" : "EUR",
10     "total" : 1529.17
11 },
12
13 /* 3 */
14 {
15     "_id" : "USD",
16     "total" : 1039.78

```

7. Calculul numarului total de cheltuieli din fiecare categorie:

```

db.MonitorCheltuieli.aggregate([
  {
    $group: {
      _id: "$categoria",
      numarCheltuieli: { $sum: 1 }
    }
  }
]);

```

```

MonitorCheltuieli 0.168 s 5 Docs
1 /* 1 */
2 {
3     "_id" : "Educatie",
4     "numarCheltuieli" : 4
5 },
6
7 /* 2 */
8 {
9     "_id" : "Divertisment",
10    "numarCheltuieli" : 3
11 },
12
13 /* 3 */
14 {
15     "_id" : "Utilitati",
16     "numarCheltuieli" : 3

```

8. Numararea cheltuielilor din fiecare locatie cu mai mult de 5 inregistrari:

```

db.MonitorCheltuieli.aggregate([
{
  $group: {
    _id: "$locatia",
    numarCheltuieli: { $sum: 1 }
  }
},
{
  $match: {
    numarCheltuieli: { $gt: 5 }
  }
}
]);

```

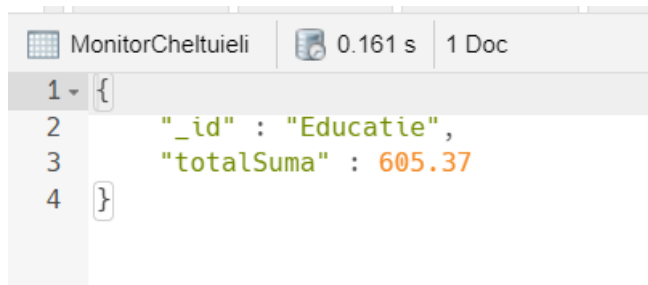
```

MonitorCheltuieli 0.184 s 1 Doc
1 {
2     "_id" : "Iasi",
3     "numarCheltuieli" : 6
4 }

```

9. Calculati suma totala a cheltuielilor pentru fiecare categorie unde exista cel putin doua documente asociate (documenteAsociate) pentru fiecare inregistrare, si in cazul in care totalul depaseste o anumita valoare.

```
db.MonitorCheltuieli.aggregate([
  {
    $match: {
      "moneda": "EUR",
      "documenteAsociate": { $exists: true }
    }
  },
  {
    $addFields: {
      numarDocumenteAsociate: { $size: "$documenteAsociate" }
    }
  },
  {
    $match: {
      numarDocumenteAsociate: { $gte: 2 }
    }
  },
  {
    $group: {
      _id: "$categoria",
      totalSuma: { $sum: "$suma" }
    }
  },
  {
    $match: {
      totalSuma: { $gt: 300 }
    }
  }
]);
```



- Primul \$match selecteaza documentele cu moneda "EUR" si unde documenteAsociate exista.
- \$addFields este folosit pentru a adauga un nou camp, numarDocumenteAsociate, care va contine numarul de elemente din array-ul documenteAsociate.
- Al doilea \$match filtreaza documentele care au cel putin doua documente asociate.
- \$group grupeaza rezultatele pe categorii si calculeaza suma totala a cheltuielilor pentru fiecare categorie.
- Ultimul \$match filtreaza grupurile care au o suma totala mai mare decat valoareaDorita.

10. Sa se afiseze detaliile fiecarui document in parte pentru cheltuielile din categoria "Transport", includand suma, moneda, data, locatia si fiecare document asociat in parte. In plus, sa se adauge un camp care arata daca suma este mai mare decat o anumita valoare (de exemplu, 100 unitati).

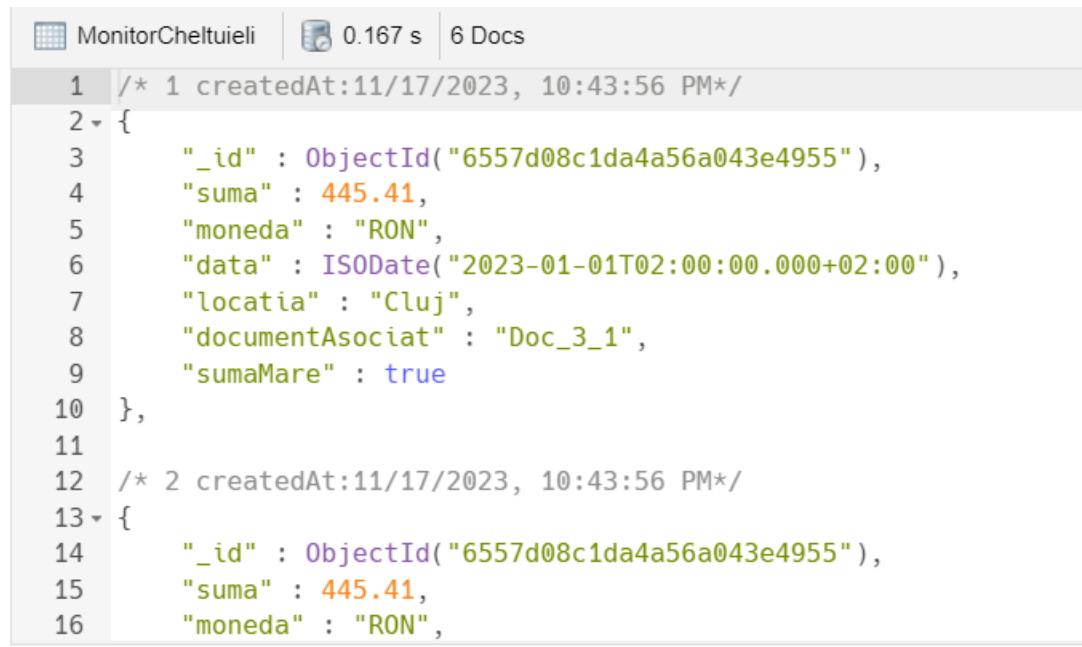
Pentru aceasta, vom folosi \$unwind pentru a descompune array-ul documenteAsociate, astfel incat fiecare element al acestuia sa fie prezentat intr-un document separat. Apoi, vom folosi \$project pentru a afisa detaliile dorite si pentru a adauga campul suplimentar care indica daca suma depaseste valoarea specificata. Iata cum ar arata query-ul:

```
db.MonitorCheltuieli.aggregate([
  {
```

```

$match: {
  "categoria": "Transport"
}
},
{
  $unwind: "$documenteAsociate"
},
{
  $project: {
    "suma": 1,
    "moneda": 1,
    "data": 1,
    "locatia": 1,
    "documentAsociat": "$documenteAsociate",
    "sumaMare": { $gt: ["$suma", 100] }
  }
}
});

```



```

MonitorCheltuieli 0.167 s 6 Docs
1 /* 1 createdAt:11/17/2023, 10:43:56 PM*/
2 {
3   "_id" : ObjectId("6557d08c1da4a56a043e4955"),
4   "suma" : 445.41,
5   "moneda" : "RON",
6   "data" : ISODate("2023-01-01T02:00:00.000+02:00"),
7   "locatia" : "Cluj",
8   "documentAsociat" : "Doc_3_1",
9   "sumaMare" : true
10 },
11
12 /* 2 createdAt:11/17/2023, 10:43:56 PM*/
13 {
14   "_id" : ObjectId("6557d08c1da4a56a043e4955"),
15   "suma" : 445.41,
16   "moneda" : "RON",

```

11. Calculul de cheltuieli mari în categoria 'Transport' si documente asociate cu o valoare mare

Acest flux de agregare va separa cheltuielile mari din categoria 'Transport' si va evidentia documentele asociate cu sume mari:

```
db.MonitorCheltuieli.aggregate([
{
  $facet: {
    "CheltuieliMariTransport": [
      { $match: { "categoria": "Transport", "suma": { $gt: 200 } } },
      { $project: { "suma": 1, "data": 1, "locatia": 1 } }
    ],
    "DocumenteAsociateCuSumaMare": [
      { $unwind: "$documenteAsociate" },
      { $match: { "suma": { $gt: 200 } } },
      { $project: { "documentAsociat": "$documenteAsociate" } }
    ]
  }
}
]);
```

MonitorCheltuieli	0.210 s	1 Doc
1	{	
2	"CheltuieliMariTransport" : [	
3	{	
4	"_id" : ObjectId("6557d08c1da4a56a043e4955"),	
5	"suma" : 445.41,	
6	"data" : ISODate("2023-01-01T02:00:00.000+02:00"),	
7	"locatia" : "Cluj"	
8	}	
9	],	
10	"DocumenteAsociateCuSumaMare" : [	
11	{	
12	"_id" : ObjectId("6557d08c1da4a56a043e4953"),	
13	"documentAsociat" : "Doc_1_1"	
14	},	
15	{	
16	"_id" : ObjectId("6557d08c1da4a56a043e4953"),	

12. Dorim sa cream bucket-uri (galeti) pentru gruparea cheltuielilor in functie de suma acestora pentru fiecare locatie. Vom imparti cheltuielile in trei categorii distincte: "Mici" (sub 100 unitati), "Medii" (intre 100 si 500 unitati) si "Mari" (peste 500 unitati).

```
db.MonitorCheltuieli.aggregate([
  {
    $bucket: {
      groupBy: "$suma",
      boundaries: [0, 100, 500, Infinity],
      default: "Alte Valori",
      output: {
        "numarCheltuieli": { $sum: 1 },
        "cheltuieli": {
          $push: {
            "locatia": "$locatia",
            "suma": "$suma",
            "moneda": "$moneda",
            "data": "$data"
          }
        }
      }
    }
  }
]);
```

```
MonitorCheltuieli 0.174 s 1 Doc
1 {
2   "_id" : 100,
3   "numarCheltuieli" : 15,
4   "cheltuieli" : [
5     {
6       "locatia" : "Timisoara",
7       "suma" : 248.26,
8       "moneda" : "EUR",
9       "data" : ISODate( "2023-01-03T02:00:00.000+02:00" )
10    },
11    {
12      "locatia" : "Iasi",
13      "suma" : 492.38,
14      "moneda" : "USD",
15      "data" : ISODate( "2023-01-24T02:00:00.000+02:00" )
16    }
17  ]
18 }
```

- \$bucket este folosit pentru a imparti documentele in categorii pe baza valorii campului suma.
- groupBy: "\$suma" indica faptul ca gruparea se face pe baza sumei cheltuielilor.
- boundaries: [0, 100, 500, Infinity] defineste limitele pentru fiecare categorie de sume: 0-100, 100-500 si peste 500.
- default: "Alte Valori" este utilizat pentru a clasifica documentele care nu se incadreaza in niciuna dintre categoriile definite.
- output specifica structura fiecarui bucket, numarand documentele (numarCheltuieli) si adunand detaliile cheltuielilor in cheltuieli.